Programme Specification (Undergraduate) FOR ENTRY YEAR: 2025/26

Date created: n/a **Last amended:** 25/04/2025 **Version no.** 1

1. Programme title(s) and code(s):

BSc Software Engineering (G600)

BSc Software Engineering with a Year Abroad (G601)

BSc Software Engineering with a Year in Industry (G602)

BSc Software Systems*

BSc Software Systems with a Year Abroad*

BSc Software Systems with a Year in Industry*

Diploma of Higher Education in Software Engineering*

Certificate of Higher Education in Software Engineering*

Notes

* An award marked with an asterisk is only available as an exit award and is not available for students to register onto and is not accredited by the British Computer Society.

a) HECOS Code

HECOS Code	%
100374	100%

b) UCAS Code (where required)

Codes listed above.

2. Awarding body or institution:

University of Leicester

3. a) Mode of study

Full-time

b) Type of study

Campus-based

4. Registration periods:

BSc Software Engineering

The normal period of registration is 3 years

The maximum period of registration 5 years

BSc Software Engineering with a Year Abroad

The normal period of registration is 4 years

The maximum period of registration 6 years

BSc Software Engineering with a Year in Industry

The normal period of registration is 4 years

The maximum period of registration 6 years

5. Typical entry requirements

A level: BBB or points equivalent from best three A levels.

BTEC Diploma: D*D*D* in appropriate subject area. BTEC must be in IT, Science or Engineering.

For Foundation Year Variant:

A level: BBC or points equivalent from best three A levels. Typically in subjects outside of the 'usual' A levels expected by the department.

BTEC Diploma: D*DM in appropriate subject area.

6. Accreditation of Prior Learning

APL will not be accepted for exemptions from individual modules, however may be considered for direct entry to year 2, on a case by case basis and subject to the general provisions of the University APL policy.

7. Programme aims

The programme aims to:

- Provide students with a state-of-the-art education in Software Engineering that includes foundations, comprehensive cover of practical applications, and a range of topics in business and management that underpin the subject.
- Provide opportunities for students to learn a wide range of engineering skills in the analysis, specification, design, implementation, testing, maintenance and documentation of computer software systems.
- Enable students to become proficient in a variety of modern programming languages, and the underlying principles of programming paradigms (concurrent, imperative, mobile, object oriented and so on).
- Enable students to explain core subjects such as algorithms, computer architecture, databases, web & mobile computing, deployment & maintenance of software, legacy systems, together with a further range of advanced subjects such as data analytics, big data, technology innovation, and Entrepreneurial Projects that reflect the expertise of the Department.
- *Skills labelled ^ are taught to a high level of insight and complexity:* Enable students to develop skills such as Communication, Teamwork^, Leadership & Supervision, Researching & Analysing^, Problem Solving & Decision Making^, Planning & Organization^; Learning, Improving & Achieving; Resilience, Adaptability & Drive; and Digital Skills^. Skills labelled ^ are taught to a high level of insight and complexity.
- Provide students with experience of both team-based and individual project work.
- To develop an appreciation for computational, mathematical and scientific thinking, along with an appreciation of the necessity for sound subject foundations, which will provide a lifelong support for careers.
- Ensure students will have expertise and understanding at a level where they can embark upon a high quality taught Masters Programme in Software Engineering.

In addition to these aims, BSc Software Engineering with a Year Abroad aims to:

- Enable students to experience modern Software Engineering from an international perspective.

- Develop students' working knowledge of a language other than English.
- Provide students with an environment that will encourage a thoughtful and mature approach to all aspects of study and life, creating graduates with broad experiences and horizons.

In addition to these aims, BSc Software Engineering with a Year in Industry, and BSc Software Engineering for Business aim to:

- Enable students to take up industrial placements where they can gain first-hand experience of the requirements, challenges and opportunities of the computing industry in the UK.
- Enable students to use and further develop the knowledge and skills gained during the first two years of the degree programme.

For Foundation Year variant, see Foundation Year Programme Specification

8. Reference points used to inform the programme specification

- QAA Benchmarking Statement for Computing
- BCS HEI Accreditation Guidelines
- Tech Partnership (formerly e-skills) Endorsement Guide
- Framework for Higher Education Qualifications (FHEQ)
- UK Quality Code for Higher Education
- University Education Strategy
- University Assessment Strategy
- University of Leicester Periodic Developmental Review Report
- External Examiners' reports (annual)
- United Nations Education for Sustainable Development Goals
- Student Destinations Data

9. Programme Outcomes

Unless otherwise stated, programme outcomes apply to all awards specified in 1. Programme title(s).

a) Discipline specific knowledge and competencies

i) Mastery of an appropriate body of knowledge

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Explain and discuss both foundations and applications of Software Engineering, together with concomitant scientific knowledge and concepts from logic and mathematics.	Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.
2. Explain, discuss and apply engineering principles and relevant scientific principles.	As above.	As above.
3. Demonstrate mastery of the core of an appropriate foreign language.	Lectures, language laboratories and learning abroad.	Assessment at host institution.

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
4. Demonstrate understanding of the core elements of industrial practice and organisation.	Work placement.	Placement Report; presentation.

ii) Understanding and application of key concepts and techniques

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Apply knowledge of Mathematics, Logic and Software Engineering to solve individual problems, both seen and unseen.	Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.
2. Apply the concepts and techniques of abstraction, reification, logical structure and modelling, that pervade Software Engineering and Software Engineering to specify, design, implement and test small to medium size computer systems.	As above.	As above.
3. Explain and apply the theoretical principles, and practical tools of Mathematics, Logic, Software Engineering, and Software Engineering, together with suitable processes and methodologies, to determine strategies for innovative solutions of large scale problems.	As above, with emphasis on all forms of project work.	As above, with emphasis on project assessments.
4. Apply sound business and management techniques to the processes of software engineering, and the deployment and maintenance of software. Analyse legacy code and synthesize innovative future software solutions.	As above	As above
5. Demonstrate ability to communicate some aspects of Software Engineering in a foreign language.	Lectures and language instruction.	As above

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
6. Work as a computing engineer in an industrial or commercial setting.	Placement Year	University report.

iii) Critical analysis of key issues

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Analyse client/customer problems, requirements and criteria, and hence plan an appropriate yet innovative solution strategy.	Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.
2. Explain and analyse the constraints of budgets, data, time, staffing and resources in the practical computing domain, undertaking suitable research. Ensure software solutions are fit-for-purpose. Manage the complete software engineering process and evaluate the end product, and to work with associated uncertainties.	As above.	As above.
3. Be able to recognise risks in the deployment and use of software systems.	As above.	As above.

iv) Clear and concise presentation of material

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
Present information in a variety of forms, chosen to maximise reader/audience impact and understanding, such as reports, dissertations, seminars, posters, blogs, podcasts, videos and other current media technologies.	Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.

v) Critical appraisal of evidence with appropriate insight

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Evaluate and appraise software systems, in terms of attributes and tradeoffs. Identify risks and safety concerns.	Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.
2. Perform software testing, and critically evaluate and analyse test results. Evaluate whether a system meets requirements, for future and for current use. Discuss legacy code and be able to construct future innovations.	As above.	As above.
3. Use relevant knowledge to appraise the commercial use and economic and long-term viability of computer systems.	As above.	As above.

vi) Other discipline specific competencies

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Explain and discuss social, legal and ethical issues as required by computing professionals. Adopt and implement professional and legal practice.	Lectures, tutorials, computer laboratories, audios & videos, group discussions, project work, guided independent study. Also background reading and research.	Written examinations, summative and formative coursework, group and individual project presentations, individual project oral examinations and project dissertations.
2. Explain and react to the rapidity of change in Software Engineering. Discuss and analyse all forms of legacy (such as legacy code). Formulate innovative and creative ideas for future advances.	As above.	As above.
3. Collect, work with and analyse all forms of data.	As above.	As above.
4. Deploy and maintain software in a commercial environment.	As above.	As above.

b) Transferable skills

i) Oral communication

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Respond to technical questions with accurate and concise answers.	Lectures and tutorials. Project supervisions.	Group and individual project presentations, individual project oral examinations.
2. Demonstrate fluent and sustained scientific, technical and business communication.	As above.	As above.
3. Demonstrate core oral communication skills in a foreign language (G401).	Language tuition.	Host University assessment.

ii) Written communication

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Write concise and accurate summaries of computing and scientific knowledge, and solutions to problems, in a variety of different formats.	Lectures, tutorials, computer laboratories, project work.	Written examinations, assessed coursework.
2. Produce properly structured, clear, advanced technical reports or dissertations.	Lectures and tutorials. Discussed in both group and individual project supervisions.	Group project assessed coursework and individual project reports.
3. Demonstrate core written communication skills in a foreign language (G401).	Lectures, tutorials, language laboratory work.	University report.

iii) Information technology

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Use a very broad range of software and IT tools, and to choose these appropriately for uses throughout Software Engineering.	Lectures, tutorials and laboratories.	Assessed (laboratory) coursework.
2. Adapt to future programming languages and paradigms, and all varieties of software tools and technology.	As above.	As above.

iv) Numeracy

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
Demonstrate understanding of the concept of number. Solve numerical problems.	Lectures, tutorials, computer laboratories.	Written examinations, assessed coursework.
Use analytical, quantitative, and graphical methods, and deploy elementary statistics.	As above, together with project work.	As above, along with group and individual project presentations and reports.

v) Team working

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
Work effectively as part of a team, organise roles and manage time, undertake assigned tasks, and ensure final completion of a team project. Identify strengths and weaknesses of team members.	Lectures, tutorials and project supervision.	Group project assessed coursework and presentations. Mini projects.

vi) Problem solving

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Solve a variety of short problems through the integration of knowledge of mathematics, logic, and Software Engineering.	Lectures and tutorials. Also covered in project supervisions.	Written examinations, assessed coursework, and project reports.
2. Use systematic analysis and design methods, and appropriate algorithms, to solve medium scale problems.	As above.	As above.
3. Analyse large-scale problems to produce suitable solutions with sensible economic and commercial compromises. Apply management techniques to allocate resources to projects.	As above.	Group and individual project presentations and reports.

vii) Information handling

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
Conduct significant background research and literature surveys, and summarise content from information sources.	Taught in lectures. Also covered in project supervisions.	Individual project reports.
Demonstrate a broad understanding of problems and issues that arise in the location, organisation, processing and evaluation of data.	As above.	Written examinations, assessed coursework, and project reports.
Recognise the need for information, and work with fuzzy, limited and possibly contradictory information.	As above.	As above.

viii) Skills for lifelong learning

Intended Learning Outcomes	Teaching and Learning Methods	How Demonstrated?
1. Demonstrate knowledge and understanding of professional and ethical issues, and aspects of the law, in the context of Computing Professionals.	Lectures and tutorials. Also covered in project supervisions.	Written examinations, assessed coursework, and project reports.
2. Demonstrate independence and time management skills.	Project supervisions and research project work. Meeting coursework deadlines.	Project reports.
3. Design a personal work plan and be able to improve performance with a clear view of long-term professional development.	Project supervisions and research project work.	As above.

10. Progression points

This programme follows the standard Scheme of Progression set out in Senate Regulations – see the version of Senate Regulation 5 governing undergraduate programmes relevant to the year of entry.

In the case that a student does not effectively engage with the externally delivered business elements of the Entrepreneurial Project, the student should formulate a requirements for a software system that solves the key elements of the original project problem and use all independent study time to specify, design and code the software. The final dissertation should contain a short account of any business elements that were completed, and evaluate and analyse the issues in engaging with the business components of the project (in the Critical Appraisal).

The following modules have restrictions on the assessment components that can be reassessed:

- CO2302

- CO3202
- CO3204

Please refer to the module specification for full details.

For Foundation Year Variant:

For progression from the Foundation Year, refer to the Foundation Year programme specification from year of entry.

In cases where a student has failed to meet a requirement to progress he or she will be required to withdraw from the course.

British Computer Society Accreditation requires that individual projects cannot be compensated therefore these final year individual projects must be passed for a successful completion of the degree

a) Year Abroad

The Year Abroad variant of this programme is offered in accordance with the University's standard specification for year abroad programme variants.

b) Year in Industry

The Year in Industry variant of this programme is offered in accordance with the University's standard specification for year in industry programme variants.

a) Course transfers

Students can transfer between BSc Software Engineering and BSc Computer Science up until the beginning of the second semester of year 1. Course transfers are not possible after this time.

11. Criteria for award and classification

This programme follows the standard scheme of undergraduate award and classification set out in Senate Regulations – see the version of *Senate Regulation 5 governing undergraduate programmes* relevant to the year of entry.

12. Special features

Emphasis on blending long-term foundational knowledge with state-of-the-art technologies and current programming languages; a structured approach to teaching a good range of programming paradigms; Software Engineering Projects involving an external client wherever possible; Entrepreneurial Projects.

The University recognises that undertaking a work placement as part the programme of study can enhance career prospects and provide added value, and as such this programme includes a 'year in industry' variant.

By experiencing real-world scenarios and applying skills and knowledge to a professional environment, students can gain a unique insight into how their studies can be utilised in industry. This will not only showcase their abilities to future employers but will also enhance their studies upon returning to university to complete your programme.

To understand the special features for year in industry undergraduate programme variants, this programme specification should be read in conjunction with the programme specification content which can be found here. This outlines details including programme aims, support, progression and duration.

- Research-inspired Education

Students on this programme will advance through the four quadrants of the University of Leicester Research-inspired Education Framework as follows:

RiE Quadrant	Narrative
	The programme puts emphasis on blending long-term foundational knowledge with state-of-the-art research-based technologies and current programming languages. It offers a structured approach to teaching a wide range of programming paradigms and looks at a range of topics in business and management. Students will take part in Software Engineering Group Project, and in Entrepreneurial or Software Engineering Projects with a number of structured milestones.
Research-briefed Bringing staff research content into the curriculum.	*Research-briefed:* Students will be exposed to a number of programming languages and paradigms (Java, Python, …) all of which are actively being used in research, as well as industry. For each programming language, the students will be shown, and taught, its most common applications. Students will be introduced to a range of advanced subjects such as data analytics, technology innovation, and Entrepreneurial Projects that reflect the expertise of the School.
Research-based Framed enquiry for exploring existing knowledge.	*Research-based:* During computer labs, students will have an opportunity to put their problem-solving and research skills in practice by solving problems with applications to data analysis, ML, AI, and more.
Research-oriented Students critique published research content and process.	*Research-oriented:* Students will be able to search information effectively, and organise and present information in the form of an IT literature survey. Students will also have the opportunity to evaluate the outcomes of a project, including social, legal and ethical considerations.
Research-apprenticed Experiencing the research process and methods; building new knowledge.	*Research-apprenticed:* Students will work to produce complete pieces of software, be it a website, a game demo, an app, from various IT applications. Students will then have the opportunity to present their work and process, and be challenged on choices that were made throughout the project.

As part of studying at a research-intensive university, students on this programme have the following extra or co-curricular opportunities available to them to gain exposure to research culture:

The School helps organise multiple Hackathons during the academic year where the students can come together and collaboratively work or build new software. These Hackathons often have industrial partnerships and collaborators, for example IBM and Capital One. Students are informed and invited to participate in these events via emails.

Students can apply to join the DriverLeics group, which was invited to demonstrate autonomous technologies at the Royal Society Summer Science Exhibition. Successful candidates will engage in research-inspired learning activities in autonomous systems, such as robotics and autonomous vehicles. They will also have opportunities to participate in national and international competitions, such as F1Tenth, and take part in local outreach and voluntary STEM activities. Throughout term, subject specific career drop-in sessions are scheduled (and added to the students' timetable), in order for students to find out more about the subject and research specific paths in Computer Science.

Teaching on this programme will be research-informed (it draws consciously on systematic inquiry into the teaching and learning process itself) in the following way:

The School supports all staff involved in teaching to gain an accredited Higher Education teaching qualification, in which they demonstrate their use of teaching theory to support their own practice and reflect on their current teaching and continuing professional development.

All module convenors are part of teaching pods, which group similar fields together. These pods are designed to provide a forum for discussion between teaching-focussed and teaching/research staff, and as a way for more experienced staff to support others by, for example, peer observation and feedback. This provides a platform for staff to share considerations and observations of their teaching experience and obtain research-based input.

Teaching staff meet once a year for a 'Teaching Away Day', which gives the opportunity to discuss some key issues in depth with the other members within the teaching pods, and shared with everyone. This gives a chance to share ideas and experience, and to identify questions that need answers.
Additionally, staff will be paired within their teaching pods to observe each other's teaching sessions then meet to agree actions in order to participate in UoL's Peer Observation of Teaching scheme.

13. Indications of programme quality

British Computer Society Accreditation will be sought, and requires that individual projects be passed at the first attempt.

14. External Examiner(s) reports

The details of the External Examiner(s) for this programme and the most recent External Examiners' reports for this programme can be found at exampapers@Leicester [log-in required]

UNIVERSITY OF LEICESTER

Programme Specification (Undergraduate) FOR ENTRY YEAR: 2025/26

Date created: n/a **Last amended:** 25/04/2025 **Version no.** 1

Appendix 1: Programme structure (programme regulations)

The University regularly reviews its programmes and modules to ensure that they reflect the current status of the discipline and offer the best learning experience to students. On occasion, it may be necessary to alter particular aspects of a course or module.

BSc Software Engineering

Level 4/Year 1 2025/26

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	n/a	60 credits	60 credits
Optional	n/a	n/a	n/a

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Sem 1	CO1101	COMPUTING FUNDAMENTALS	15 credits
Sem 1	CO1102	PROGRAMMING FUNDAMENTALS	15 credits
Sem 1	CO1103	MATHEMATICS FUNDAMENTALS	15 credits
Sem 1	CO1104	COMPUTER ARCHITECTURE	15 credits
Sem 2	CO1105	INTRODUCTION TO OBJECT ORIENTED PROGRAMMING	15 credits
Sem 2	CO1106	REQUIREMENTS ENGINEERING AND PROFESSIONAL PRACTICE	15 credits
Sem 2	CO1107	ALGORITHMS, DATA STRUCTURES AND ADVANCED PROGRAMMING	15 credits

Delivery period	Code	Title	Credits
Sem 2	CO1109	BUSINESS AND FINANCIAL COMPUTING	15 credits

Notes
N/A

Level 5/Year 2 2026/27

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	n/a	60 credits	45 credits
Optional	n/a	n/a	15 credits

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Sem 1	CO2123	SOFTWARE ARCHITECTURE AND SYSTEM DEVELOPMENT – I	15 credits
Sem 1	CO2301	PROJECT MANAGEMENT	15 credits
Sem 1	CO2101	OPERATING SYSTEMS AND NETWORKS	15 credits
Sem 1	CO2102	DATABASES AND DOMAIN MODELLING	15 credits
Sem 2	CO2124	SOFTWARE ARCHITECTURE AND SYSTEM DEVELOPMENT – II	15 credits
Sem 2	CO2302	SOFTWARE ENGINEERING GROUP PROJECT	15 credits
Sem 2	CO2115	INFORMATION SECURITY FUNDAMENTALS	15 credits

Notes
N/A

Option modules

Delivery period	Code	Title	Credits
Semester 2	CO2106	DATA ANALYTICS	15 credits
Semester 2	CO2104	USER INTERFACES DESIGN AND EVALUATION	15 credits
Semester 2	CO2114	FOUNDATIONS OF ARTIFICIAL INTELLIGENCE	15 credits

Notes
Choose one optional module in semester 2.

Level 6/Year 3 2027/28

Credit breakdown

Status	Year long	Semester 1	Semester 2
Core	45 credits	15 credits	n/a
Optional	n/a	30 credits	30 credits

120 credits in total

Core modules

Delivery period	Code	Title	Credits
Year long	CO3202	ENTREPRENEURIAL PROJECT*	45 credits
Year long	CO3204	SOFTWARE ENGINEERING PROJECT*	45 credits
Sem 1	CO3101	COMPUTERS, SOCIETY & PROFESSIONALISM	15 credits

Notes
* Year 3 students must decide between CO3204 and CO3202 for their final-year project. Enrollment to CO3202 may be subject to an interview.

Option modules

Delivery period	Code	Title	Credits
Semester 1	CO3091	COMPUTATIONAL INTELLIGENCE AND SOFTWARE ENGINEERING	15 credits
Semester 1	CO3095	SOFTWARE MEASUREMENT AND QUALITY ASSURANCE	15 credits
Semester 1	CO3102	MOBILE AND WEB APPLICATIONS	15 credits
Semester 1	CO3104	COMPUTATIONAL CREATIVITY	15 credits
Semester 1	CO3105	C++ PROGRAMMING	15 credits
Semester 1	CO3219	INTERNET AND CLOUD COMPUTING	15 credits
Semester 2	CO3002	ANALYSIS AND DESIGN OF ALGORITHMS	15 credits
Semester 2	CO3093	BIG DATA AND PREDICTIVE ANALYTICS	15 credits
Semester 2	CO3099	FOUNDATIONS OF CYBER SECURITY	15 credits
Semester 2	CO3103	TECHNOLOGY MANAGEMENT	15 credits
Semester 2	CO3111	FUNCTIONAL PROGRAMMING	15 credits
Semester 2	CO3113	AI FOR SPACE	15 credits

Notes
Choose 30 credits (2 modules) of optional modules in each semester.

BSc SOFTWARE ENGINEERING WITH A YEAR ABROAD

First and Second Year Modules

As for the first- and second-year of the BSc degree in Software Engineering.

Third Year Modules

The third year will be spent abroad taking approved courses either in an institution associated with the Informatics Department via an ERASMUS bilateral agreement or in a university that has a Study Abroad exchange partnership agreement with the University of Leicester. Students will normally be required to complete the year and to reach a pass level of attainment in 60 credits of Software Engineering modules. Failure to do so will result in the student reverting to the three year BSc Software Engineering degree. The marks awarded during the year abroad do not contribute to the final degree classification.

Note: Transfer will be confirmed only after successful completion of the first year.

Fourth Year Modules

As for the third-year of the BSc degree in Software Engineering.

BSc SOFTWARE ENGINEERING WITH A YEAR IN INDUSTRY

First and Second Year Modules

As for the first- and second-year of the BSc degree in Software Engineering.

Third Year Modules

1. Students will work within a sponsoring company for one year between 1 July of the second year of the course and the start of the following year.
2. During their one-year placement students will undertake a programme of training and work experience which will be agreed by the sponsoring company and the University.
3. Students will be expected to keep a logbook recording their training and experience that is to be presented for approval to the sponsoring company and the University.
4. Students will be issued with a *Certificate of Industrial Studies* indicating successful completion of their placement.
   Students who do not satisfactorily complete their industrial placement will be transferred to the B.Sc. Software Engineering degree.

   The Year in Industry does not contribute to the final degree classification.

Fourth Year Modules

As for the third-year of the BSc degree in Software Engineering.

Appendix 2: Module specifications

See undergraduate module specification database [login required] (Note - modules are organized by year of delivery).